Caché ECP Clusters on Red Hat Enterprise Linux

Version 5.1
15 June 2006

InterSystems Corporation 1 Memorial Drive Cambridge MA 02142 www.intersystems.com

Caché ECP Clusters on Red Hat Enterprise Linux
Caché Version 5.1 15 June 2006
Copyright © 2006 InterSystems Corporation.
All rights reserved.

This book was assembled and formatted in Adobe Page Description Format (PDF) using tools and information from the following sources: Sun Microsystems, RenderX, Inc., Adobe Systems, and the World Wide Web Consortium at www.w3c.org. The primary document development tools were special-purpose XML-processing applications built by InterSystems using Caché and Java.

The Caché product and its logos are trademarks of InterSystems Corporation.

The Ensemble product and its logos are trademarks of InterSystems Corporation.

The InterSystems name and logo are trademarks of InterSystems Corporation.

This document contains trade secret and confidential information which is the property of InterSystems Corporation, One Memorial Drive, Cambridge, MA 02142, or its affiliates, and is furnished for the sole purpose of the operation and maintenance of the products of InterSystems Corporation. No part of this publication is to be used for any other purpose, and this publication is not to be reproduced, copied, disclosed, transmitted, stored in a retrieval system or translated into any human or computer language, in any form, by any means, in whole or in part, without the express prior written consent of InterSystems Corporation.

The copying, use and disposition of this document and the software programs described herein is prohibited except to the limited extent set forth in the standard software license agreement(s) of InterSystems Corporation covering such programs and related documentation. InterSystems Corporation makes no representations and warranties concerning such software programs other than those set forth in such standard software license agreement(s). In addition, the liability of InterSystems Corporation for any losses or damages relating to or arising out of the use of such software programs is limited in the manner set forth in such standard software license agreement(s).

THE FOREGOING IS A GENERAL SUMMARY OF THE RESTRICTIONS AND LIMITATIONS IMPOSED BY INTERSYSTEMS CORPORATION ON THE USE OF, AND LIABILITY ARISING FROM, ITS COMPUTER SOFTWARE. FOR COMPLETE INFORMATION REFERENCE SHOULD BE MADE TO THE STANDARD SOFTWARE LICENSE AGREEMENT(S) OF INTERSYSTEMS CORPORATION, COPIES OF WHICH WILL BE MADE AVAILABLE UPON REQUEST.

InterSystems Corporation disclaims responsibility for errors which may appear in this document, and it reserves the right, in its sole discretion and without notice, to make substitutions and modifications in the products and practices described in this document.

Caché, InterSystems Caché, Caché SQL, Caché ObjectScript, Caché Object, Ensemble, InterSystems Ensemble, Ensemble Object, and Ensemble Production are trademarks of InterSystems Corporation. All other brand or product names used herein are trademarks or registered trademarks of their respective companies or organizations.

For Support questions about any InterSystems products, contact:

InterSystems Worldwide Customer Support

Tel:	+1 617 621-0700
Fax:	+1 617 374-9391
Email:	support@InterSystems.com

Table of Contents

Caché ECP Clusters on Red Hat Enterprise Linux 1
1 Pre-installation Planning 1
2 Configuring the Cluster Services for Caché 2
2.1 Define the Caché Cluster Services 3
2.2 Install Caché 4
3 Configuring the Second Node 5
4 Adding Caché to the Cluster Services 5
4.1 Caché Initialization File for Linux 7
5 Maintaining the Caché Registry When Upgrading 9

Caché ECP Clusters on Red Hat Enterprise Linux

Caché ECP Clusters is a high availability feature that enables failover from one ECP data server to another, using operating system level clustering to detect a failed server. Caché ECP Clusters technology has been tested and is supported on Red Hat Enterprise Linux AS version 4. This document contains details of how to configure the cluster and is organized into the following sections:

- Pre-installation Planning
- Configuring the Cluster Services for Caché
- Configuring the Second Node
- Adding Caché to the Cluster Services
- Maintaining the Caché Registry When Upgrading

For detailed information on configuring the cluster on the Red Hat Enterprise Linux, see the *Red Hat Cluster Suite, Configuring and Managing a Cluster*, at the Red Hat Web site.

1 Pre-installation Planning

This section outlines the requirements for configuring the cluster system. Subsequent sections describe the steps to install, define, and configure the various components of the cluster. To plan the process for setting up the cluster system, first determine whether the configuration is a *hot-standby* configuration or an *active-active* configuration. In a hot-standby configuration only one node is running Caché at a time.

In an active-active configuration each node is running its own instance of Caché. The nodes do not have direct access to the same databases; each database is assigned to one Caché instance or the other. You can network the Caché instances with ECP and use namespace definitions to project the same data from both nodes.

An *active-active* configuration requires the following tasks:

- Assign each Caché instance a unique cluster IP address.
- Assign each Caché instance a unique default port number.
- Assign each Caché instance one or more disk partitions where it is installed and where the databases reside. Caché instances must not share partitions.

Both types of configuration require the following tasks:

- Calculate and update the settings for the various kernel parameters that need to be modified to support the Caché installation and verify that both nodes can support the requirements. If you are configuring an active-active cluster, add the parameter values together, as one node may at some point be running both instances of Caché.

  See the "Calculating System Parameters for UNIX and Linux" appendix of the *Caché Installation Guide* for more information.
- Assign a virtual IP address for each active Caché instance.
- Verify that there are enough partitions for each active Caché instance to have its own.
- Verify that the Red Hat Cluster Manager is installed and the cluster service is running. (See the "Cluster Administration" chapter of *Red Hat Cluster Suite, Configuring and Managing a Cluster* for a description of the Cluster Status Tool and the .**service** command.)
- Choose the Red Hat service names for your Caché instance names. Though not required, using the same names simplifies the cluster configuration.

2 Configuring the Cluster Services for Caché

To prepare the cluster system and configure the first cluster node, perform the following steps:

1. Use the **parted** utility to create the partitions as described in the "Partitioning Disks" section of *Red Hat Cluster Suite, Configuring and Managing a Cluster*.
2. Create the mount points for the partitions on each cluster node.
3. Choose one node on which to start work.
4. Define the Caché cluster services.
5. Install Caché on this node.

2.1 Define the Caché Cluster Services

Use the **cluadmin** utility to define the Caché cluster services. See the "Using the cluadmin Utility" section of *The Red Hat Cluster Manager Installation and Administration Guide* for more information.

Starting and Stopping the Cluster Software

1. Specify the virtual IP addresses and the storage, but not the startup script, for Caché.

   While each instance is given a preferred node, specify `no` for the **relocate** subcommand. If you specify `yes`, a service automatically relocates to the preferred node when it starts. Though it is acceptable to specify `yes` for Caché, it is preferable to specify `no` so that you can control the relocation process manually.

   Use the **service relocate** command to move services between nodes. See the "Relocating a Service" section of *The Red Hat Cluster Manager Installation and Administration Guide* for more information.

   The following output from the **service show config** command shows the configuration for a service named `cacheha1` with a virtual IP address of `192.9.202.197` and one disk partition assigned:

```
cluadmin>service show config cacheha1

name: cacheha1                          -->Name picked for this service
preferred node: lx4
relocate: no
user script: None                       -->Did not specify this yet
monitor interval: 0                     -->Specify 0 here
IP address 0: 192.9.202.197             -->Virtual IP address assigned
  netmask 0: 255.255.0.0
  broadcast 0: 192.9.255.255
device 0: /dev/sdc4                     -->Partition name
  mount point, device 0: /storage2      -->Mount point for the partition
  mount fstype, device 0: ext3
  mount options, device 0: rw
  force unmount, device 0: yes
  samba share, device 0: None
```

2. Define all necessary services and assign their storage.

3. Use the **service show state** command to list the services and their current states. If any services are disabled, enable them with the **service enable** command. If any are running on the other node, move them to the current node with the **service relocate** command.

See the "Service Configuration and Administration" chapter of *The Red Hat Cluster Manager Installation and Administration Guide* for more information.

2.2 Install Caché

Install Caché on the first node. When the installation process asks for an instance name, use the same name as you used for the cluster service.

When the installation completes, use the System Management Portal to make the following configuration changes. If you are installing multiple instances of Caché, do this after each installation, not when they are all complete.

1. *Verify unique port numbers* — The installation procedure assigns unique port numbers to multiple instances on the same machine. From the **[Home] > [Configuration] > [Memory and Startup]** page, verify the instance has a unique **SuperServer Port Number**.

2. *Update license managers* — From the **[Home] > [Licensing]** page, click **License Server**.

   Click **Edit** in the local server row and replace the **Name/IP Address** with the name (or IP address) of the current node. Do not use the virtual IP address; use the real IP address (or DNS name). Click **Save**.

   Add a second license manager (click **Add**) and enter the IP address or DNS name of the other cluster member. Again, use the real name or IP address, not the virtual IP address for the cluster. Click **Save**.

3. *Configure ECP* — If you are setting up an active-active cluster and using ECP between the instances of Caché, you may configure that now or wait until later. Navigate to the **[Home] > [Configuration] > [ECP Settings]** page. Click **Add Remote Data Server** and define the other instance of Caché as a server to this one. For clarity, choose the cluster instance name for the name of the ECP data server. The portal **Create a Database** wizard uses this name to refer to the remote node. Use the *virtual IP address* that you assigned to that instance for the **Host Name**, not the real IP address or the real DNS name.

4. *Increase maximum number of ECP data and application servers* — If there are other ECP data servers or application servers in your network, increase the maximum settings as appropriate for your system. Navigate to the **[Home] > [Configuration] > [ECP Settings]** page. Under **This System as an ECP Data Server**, increase **Maximum number of application servers** and under **This System as an ECP Application Server**, increase **Maximum number of data servers**. Enable the ECP service if it is disabled.

See the "Configuring Distributed Systems" chapter of the *Caché Distributed Data Management Guide* for more information on configuring ECP.

Stop this running instance of Caché and repeat the process to install and configure all additional instances of Caché on this node.

The first cluster node is configured. Before continuing, invoke **ccontrol list** or **ccontrol all** at a shell prompt to gather the information necessary for configuring the second node.

3 Configuring the Second Node

Use the **service relocate** command of **cluadmin** to move the services (the storage you defined) to the other cluster node.

Define the Caché instances on this node in one of two ways:

- Run the installation procedure again and give the same instance name you used on the other node. Install into the same directory.

  This is the best option if you are configuring a Web server for use with Caché, because the Web server configuration is "local" to each node. Reinstalling Caché does not affect the changes you made to the configuration (cache.cpf) file.

- If you do not need Caché to configure the Web server on the other node, register the Caché instances on the local node using the **ccontrol create** command:

  ```
  ccontrol create $cfgname directory=$tgtdir versionid=$ver
  ```

  For example, if **ccontrol all** on the first node displays:

  ```
      Configuration   Version ID       Port   Directory
     --------------  ---------------- -----  ----------
  dn  CACHEHA1        5.1.0.900         1973  /store1/c51ha
  db  CACHEHA2        5.1.0.900         1972  /store2/c51ha2
  ```

  Run:

  ```
  ccontrol create cacheha1 directory="/store1/c51ha" versionid="5.1.0.900"
  ccontrol create cacheha2 directory="/store2/c51ha2" versionid="5.1.0.900"
  ```

Use **ccontrol start** to start the instances and verify that they work, shut them down with **ccontrol stop**.

4 Adding Caché to the Cluster Services

The scripts necessary to start and stop Caché as part of service failover do not ship with Caché. An example of the main script is included in the *Caché Initialization File for Linux*

section. After creating the main initialization script, perform the following steps to add Caché to the cluster services:

1. Create a script in /etc/rc.d/init.d for each instance of Caché you install—one for each Caché cluster service you define. Model it after one of the two following examples:

   Example 1:

   ```
   #!/bin/ksh
   /etc/rc.d/init.d/cache $1 cacheha1 failover
   exit ?$
   ```

   Example 2:

   ```
   #!/bin/ksh
   /usr/local/etc/cachesys/cache-init $1 cacheha1 failover
   exit ?$
   ```

   Replace `cacheha1` with your instance name. Name the script in the form "/etc/rc.d/init.d/cache-<*inst*>" where <*inst*> is the instance name (for this example, the file is: /etc/rc.d/init.d/cache-cacheha1).

2. Use the **service modify** command of the **cluadmin** utility to update the script location for each service. Leave the monitor interval set to `0`.

   The script supports the **status** command; however, if Caché becomes unresponsive and has a non-zero monitor interval, it fails over automatically to the other cluster member which may prevent it from collecting any information required to diagnose the problem.

3. Relocate the services to the node on which you want them to start. If it is the currently active node (that is, before adding the script, the services were controlling the storage availability on this node), use the **service disable** and **service enable** commands. Otherwise, use the **service relocate** command.

   The **service relocate** and **service disable** commands call the script with the *stop* parameter, which attempts to shut down Caché gracefully.

Caché is now part of the failover cluster services. Test the cluster using the following procedure:

1. Unplug one node.
2. Verify that the version of Caché that was running on the stopped node starts on the second node.
3. Turn the failed machine back on; Caché should remain running on the second node (it should not fail back automatically).
4. Unplug the second node; both Caché instances should migrate to the first node.

5. Turn the second node back on.
6. Use the **service relocate** command to move one of the instances of Caché back to the second node.
7. Try to connect the System Management Portal to the two instances of Caché using the cluster virtual IP addresses. Look at the server and instance names in the title bar to determine which node you have connected to and click **About** to see the path name to the cache.cpf file.

4.1 Caché Initialization File for Linux

This section shows a sample script to start and stop Caché on the Red Hat Enterprise Linux AS. Save the file as /etc/rc.d/init.c/cache or /usr/local/etc/cachesys/cache-init and set the protection to 755. The sample script:

```
#!/bin/ksh
#   cache
#
#   Cache "System V init" script for Linux systems
#
#   Copyright (c) 2003 - 2006 by InterSystems.
#   Cambridge, Massachusetts, U.S.A.  All rights reserved.
#   Confidential, unpublished property of InterSystems.
# ------------------------------------------------------------------
#   This script is put in the init.d directory and is used by
#   the HA failover package to start a Cache instance when
#   the node that was "serving" it failed.
#
#   Three arguments should be specified:
#      hacache start <inst name> failover
#   where <inst name> is the name of the instance that
#   is displayed by "/usr/bin/ccontrol all" in the 2nd column.
#
#   This script can be used to start Cache if Cache is currently
#   down (meaning it is down on both nodes).
#
#   On failover, it removes the .ids file. If it is not in failover,
#   it tries to restart Cache anyway. If the node trying to start
#   Cache was the node it was last running on (for example, the cluster
#   is rebooting) then it will succeed. If not, it will fail.
#
#   When it fails it could be because Cache is running on the other node
#   or it could be that it was running on the other node when the cluster
#   crashed and now we're attempting startup on a different node,
#   in which case the .ids file either needs to be manually removed or the
#   service needs to be started on the other node.
#
#   It is very dangerous to call this script and specify the failover
#   flag outside of the failover scripts. In an HA environment where
#   multiple nodes can see the attached storage simultaneously (eg. NFS
#   mounted file systems) it is possible to start Cache from the same
#   directory on both nodes; Cache does not currently prevent this.
#   If this occurs the results will be disasterous and both nodes will
#   have to be shut down, database degradation may need to be repaired,
#   and so on.
```

```
#
if [ "$2" = "" ]
then
  type="xxxx"  #invalid option, forces usage message
else
  inst=$2  #cache instance to play with
  state=$3   #failover or "nothing"
#
  basdir=`/usr/bin/ccontrol list $inst | grep -i directory | awk {'print $2'}`
  localnode=`uname -a | awk '{print $2}'`
  if [ "${basdir}" = "" ]
  then
  echo "Instance $inst not found"
  exit 1
  fi
  type=$1
fi
#
#See how we were called.
case "$type" in
  (start)
        # Start daemons.
   if [[((-e ${basdir}/mgr/cache.ids) &&("${state}" == "failover" ))]]
   then
      echo "Removing $basdir/mgr/cache.ids during HA failover"
      rm -f $basdir/mgr/cache.ids
   fi
   echo "Starting Cache-HA inst $inst on $localnode"
   ccontrol start $inst quietly
   status=$?
   case $status in
   (1)
       echo "...Failed to start"
       exit 1
       ;;
   (0)
       echo "...Started"
       exit 0
   esac
        ;;
  (stop)
        # Stop daemons.
        echo "Stopping Cache-HA inst $inst on $localnode"
   ccontrol stop $inst quietly
   status=$?
   case $status in
   (1)
       echo "Cache instance $inst failed to stop"
       exit 1
       ;;
   (0)
       echo "Cache instance $inst stopped"
       exit 0
   esac
   ;;
  (status)
   FIELDWIDTH=2
   state=`/usr/bin/ccontrol all | grep -i $inst | awk {'print $1'}`
   if [ "$state" = "up" ]
   then
      exit 0   #cache is up
   fi
   exit 1   #cache is down or we can't tell
        ;;
  (restart)
        $0 stop $2 $3 || :
        $0 start $2 $3
        ;;
```

```
  (*)
        echo "Usage: $0 {start|stop|status|restart} <inst> [failover|null]"
        exit 1
esac

exit 0
```

5 Maintaining the Caché Registry When Upgrading

You can upgrade Caché in a failover cluster with Caché running on either cluster member. However, the registry which Caché maintains (displayed with **ccontrol all** and **ccontrol list**) does not display the correct version id on the node that did not run the upgrade. Update this manually using the **ccontrol update** command. The syntax is:

```
ccontrol update $instname versionid=$ver
```

For example, to set the current version id to `5.1.0.900` for instance `cacheha1`, run:

```
ccontrol update cacheha1 versionid="5.1.0.900"
```